PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-063517
(43)Date of publication of application : 06.03.1998

(51)Int.Cl. G06F 9/46

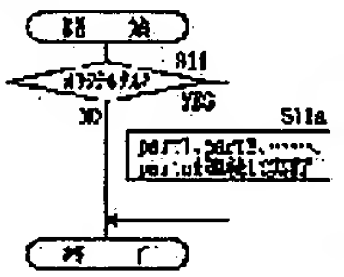
(21)Application number : 08-239774
(22)Date of filing : 22.08.1996

(71)Applicant : NIPPON TELEGR & TELEPH CORP <NTT>
(72)Inventor : NAGANUMA JIRO
ENDO MAKOTO

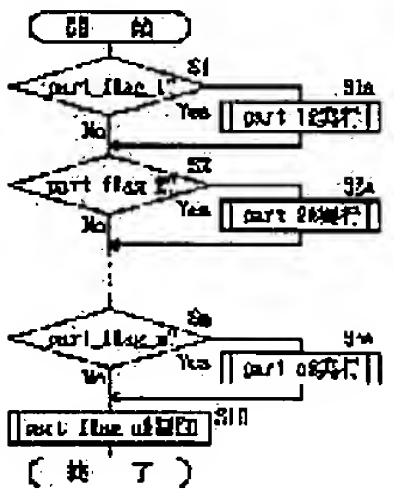
(54) REAL TIME INFORMATION PROCESSING METHOD AND ITS DEVICE

(57)Abstract:
PROBLEM TO BE SOLVED: To provide real time information processing method and its device capable of excluding a hardware for controlling the priority of multi-interruption which is essential to a conventional example, a memory for a context switch necessary for switching an original task and the overhead of processing time, etc., in real time processing.
SOLUTION: At the time of processing plural original tasks on single CPU by time-multiplexing, each of the original tasks are divided to plural sub-tasks smaller than the original task of its own to select one of these divided plural subtasks to execute. Then in order that the number of the starting times of a basic polling loop which starts the subtasks once for each original task may be at least once within a prescribed and fixed time, the division is executed.

(1)



(2)



F-99EC0439

P-725

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号

特開平10－63517

(43)公開日　平成10年(1998)３月６日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　9/46	３４０		Ｇ０６Ｆ　9/46	３４０Ｅ ３４０Ｂ

審査請求　未請求　請求項の数６　ＦＤ　（全 12 頁）

(21)出願番号　特願平8－239774

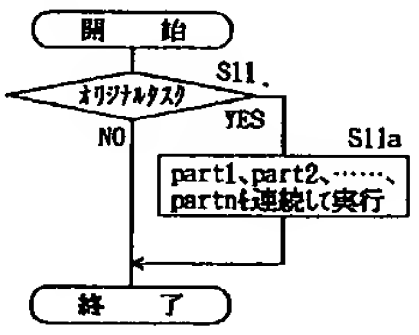
(22)出願日　平成８年(1996)８月22日

(71)出願人　000004226
日本電信電話株式会社
東京都新宿区西新宿三丁目19番２号
(72)発明者　長沼　次郎
東京都新宿区西新宿三丁目19番２号　日本電信電話株式会社内
(72)発明者　遠藤　真
東京都新宿区西新宿三丁目19番２号　日本電信電話株式会社内
(74)代理人　弁理士　川久保　新一

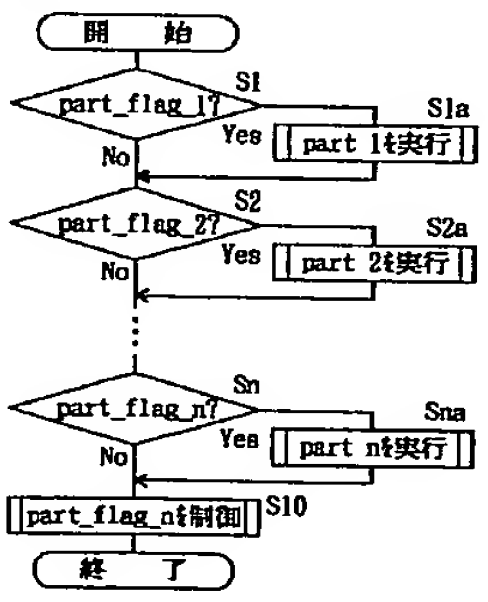
(54)【発明の名称】　リアルタイム情報処理方法およびその装置

(57)【要約】
【課題】　リアルタイム情報処理において、従来例に不可欠な多重割り込みのプライオリティ制御用のハードウェアと、オリジナルタスク切替えに必要なコンテキストスイッチ用のメモリと、処理時間と等のオーバヘッドを排除することができるリルタイム情報処理方法および装置を提供することを目的とするものである。
【解決手段】　複数のオリジナルタスクを、１つのＣＰＵ上で時間多重で処理する場合、上記各オリジナルタスクを、自らのオリジナルタスクよりも小さい複数のサブタスクに分割し、この分割された複数のサブタスクからその１つを選択し、実行するものであり、各オリジナルタスク毎に上記サブタスクを１回づつ起動する基本ポーリングループの起動回数が、所定の一定時間以内に少なくとも１回になるように、上記分割が行われるものである。

(1)



(2)



【特許請求の範囲】
【請求項１】　複数のオリジナルタスクを、１つのＣＰＵ上で時間多重で処理するリアルタイム情報処理方法において、
上記各オリジナルタスクを、自らのオリジナルタスクよりも小さい複数のサブタスクに分割するオリジナルタスク分割段階と；上記分割された複数のサブタスクからその１つを選択し、実行するサブタスク選択・実行段階と；を有し、上記各オリジナルタスク毎に上記サブタスクを１回づつ起動する基本ポーリングループの起動回数が、所定の一定時間以内に少なくとも１回になるように、上記分割が行われ、所定のリアルタイム情報処理を実現することを特徴とするリアルタイム情報処理方法。
【請求項２】１つのＣＰＵと専用ハードウェアとで構成され、上記ＣＰＵと上記専用ハードウェアとの間で情報の授受を行いながら処理を進める情報処理装置によって、複数のオリジナルタスクを、上記ＣＰＵ上で時間多重で処理するリアルタイム情報処理方法において、
上記各オリジナルタスクを、自らのオリジナルタスクよりも小さい複数のサブタスクに分割するオリジナルタスク分割段階と；上記分割された複数のサブタスクからその１つを選択し、実行するサブタスク選択・実行段階と；を有し、上記各オリジナルタスク毎に上記サブタスクを１回づつ起動する基本ポーリングループの起動回数が、所定の一定時間以内に少なくとも１回になるように、上記分割が行われ、所定のリアルタイム情報処理を実現することを特徴とするリアルタイム情報処理方法。
【請求項３】　複数のオリジナルタスクを、１つのＣＰＵ上で時間多重で処理するリアルタイム情報処理装置において、
上記各オリジナルタスクを、自らのオリジナルタスクよりも小さい複数のサブタスクに分割するオリジナルタスク分割手段と；上記分割された複数のサブタスクからその１つを選択し、実行するサブタスク選択・実行手段と；を有し、上記各オリジナルタスク毎に上記サブタスクを１回づつ起動する基本ポーリングループの起動回数が、所定の一定時間以内に少なくとも１回になるように、上記分割が行われ、所定のリアルタイム情報処理を実現することを特徴とするリアルタイム情報処理装置。
【請求項４】１つのＣＰＵと専用ハードウェアとで構成され、上記ＣＰＵと上記専用ハードウェアとの間で情報の授受を行いながら処理を進める情報処理装置によって、複数のオリジナルタスクを、上記ＣＰＵ上で時間多重で処理するリアルタイム情報処理装置において、
上記各オリジナルタスクを、自らのオリジナルタスクよりも小さい複数のサブタスクに分割するオリジナルタスク分割手段と；上記分割された複数のサブタスクからその１つを選択し、実行するサブタスク選択・実行手段と；を有し、上記各オリジナルタスク毎に上記サブタスクを１回づつ起動する基本ポーリングループの起動回数が、所定の一定時間以内に少なくとも１回になるように、上記分割が行われ、所定のリアルタイム情報処理を実現することを特徴とするリアルタイム情報処理装置。
イム情報処理装置。
【請求項５】　請求項３または請求項４において、上記リアルタイム情報処理装置は、１つのチップ上に構成され、所定のリアルタイム情報処理を実現することを特徴とするリアルタイム情報処理装置。
【請求項６】　請求項３～請求項５のいずれか１項において、
上記所定のリアルタイム情報処理は、複数の入力ポートから入力された複数の入力ストリーム中のデータを、所定のシンタックスに従って、データの組立、分解、挿入、削除、加工、並び替え等の処理を行い、複数の出力ポートへ複数の出力ストリームとして出力するプロトコル処理であることを特徴とするリアルタイム情報処理装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、リアルタイム情報処理において、従来不可欠であった多重割り込みのプライオリティ制御用のハードウェアと、オリジナルタスク切替えに必要なコンテキストスイッチ用のメモリと、処理時間と等のオーバヘッドを排除するリアルタイム情報処理に関するものである。
【０００２】また、本発明は、ＣＰＵと専用ハードウェアとの間で情報の授受を行いながら処理を進める情報処理装置（以下、「エンベディドシステム」と呼ぶ）上に、または、それらを１つのチップ上に集積したオンチップ・エンベディドシステム(On Chip Embedded System) 上に構築するリアルタイム情報処理に関するものである。
【０００３】
【従来の技術】従来、複数のオリジナルタスク（処理の主体）を、１つのＣＰＵ（計算機）上で時間多重で処理を行い、しかも、所定の一定時間以内に全てのオリジナルタスクの起動（呼び出し）を保証するリアルタイム情報処理方法として、下記文献［１］が示す方法を採るのが一般的である。また、１つのＣＰＵと専用ハードウェアとで構成され、上記ＣＰＵと上記専用ハードウェアとの間で情報の授受を行いながら処理を進める情報処理装置において、上記ＣＰＵ上で、複数のオリジナルタスクを時間多重で処理するリアルタイム情報処理方法としても、下記文献［１］が示す方法を採るのが一般的である。
【０００４】なお、通常の計算機システムの構成として、文献［２］が示す構成が知られ、エンベディドシステムの構成として、文献［３］が示す構成が知られている。
・文献［１］：J.L.Perterson 他,"オペレーティングシ

ステムの概念", 培風館, 1987.
・文献［２］：J. L. Hennessy and D. A. Patterson, "Computer Architecture:A Quantitative Approach", Morgan Kaufmann Publishers, Inc., 1990.
・文献［３］：Daniel D. Gaiski 他, "Specification and Design of Embedded Systems", Prentice Hall, 1994.
従来のリアルタイム情報処理方法における基本的な考え方は、上記文献［１］に示されているように、オリジナルタスクの緊急度に応じて各オリジナルタスクに優先度を付与し、その優先度順に、次に実行するオリジナルタスクを起動するものである。
【０００５】この場合、現在実行中のオリジナルタスクよりも優先度の高いオリジナルタスクが起動されたときに「割り込み」が発生し、現在実行中のオリジナルタスクの処理を一旦中断し、上記起動された優先度の高いオリジナルタスクを先に実行し、この起動された優先度の高いオリジナルタスクの実行が終了した後に、上記中断していたオリジナルタスクの実行を再開する。すなわち、従来の方法は、割り込みを前提とする処理方法であり、この割り込み処理は、リアルタイム性を確保するためには不可欠な処理である。
【０００６】
【発明が解決しようとする課題】ところで、上記文献［１］に示す従来のリアルタイム情報処理方法を、たとえば、上記文献［２］に示す通常の計算機システム、または、特に、文献［３］に示すエンベディドシステム上で実現する場合、次の新たな問題が生じる。
【０００７】上記エンベディドシステム(Embedded System)は、ＣＰＵと専用ハードウェアとで構成され、ＣＰＵ上のソフトウェアと専用ハードウェアとの間で情報の授受を行いながら処理を進めるシステムであり、このようにすることによって、専用ハードウェアの「高速性」とＣＰＵ上のソフトウェアによる「柔軟性」とを兼ね備えたシステムとして注目されているシステムである。
【０００８】上記エンベディドシステムにおいて、専用ハードウェア（ＨＷ）とソフトウェア（ＳＷ）との間で行われる情報の授受は、一般に、数十個を越えるＨＷ／ＳＷインタフェースレジスタを介して行われ、これらの情報の授受は、基本的に頻繁に発生ししかも非同期的に発生する。したがって、割り込みを前提とする従来のリアルタイム情報処理においては、複数のＨＷ／ＳＷインタフェースレジスタからの多重でしかも非同期な割り込みをサポートしなければならないので、多重割り込みのプライオリティ制御を行うためには、ハードウェア規模が大きくなるという問題があり、しかも、ハードウェア全体の構成が複雑になるという問題がある。
【０００９】また、優先度が付与された各オリジナルタスクの起動、中断、再開、優先度制御等を処理するために、リアルタイムＯＳをサポートしなければならないので、頻繁に発生するオリジナルタスク切替えに必要なコンテキストスイッチ用に、膨大なメモリと処理時間とが必要であるというソフトウェアの問題がある。たとえば、次の文献［４］、［５］に示すように、最も小さく軽いリアルタイムＯＳでも、数十ｋＢ～数百ｋＢのメモリを必要とする。
・［４］M. Accetta 他, "Mach:A New Kernel Foundation for UNIX Development", Proc. Summer 1986 USENIX, pp. 93-112, 1986.
・［５］J. K. Ousterhout他, "The Sprite Network Operating Syste", IEEE Computer, Vol. 21, No. 2, pp. 23-26, 1988.
上記ハードウェアの問題とソフトウェアの問題、換言すれば、要求条件は、エンベディドシステム（通常はオンボード）としては、効率的ではないものの、現状では実現がまだ可能である。しかし、将来的なオンチップのエンベディドシステムとしては、効率的ではないばかりでなく、非現実的なものとなることが想定される。
【００１０】一方、上記割り込みを用いないで、「ＭＰＥＧ２システムパートのプロトコル処理」を実現しようとすると、本来実現しなければならないリアルタイム処理が満足されないという問題が発生する。なお、「ＭＰＥＧ２システムパート」に関しては、下記文献［６］に規定されている。
・［６］Systems-Generic Coding of Moving Pictures and Associated Audio-ISO/IEC 13818-1 International Standard, November 11, 1994.
上記ＭＰＥＧ２システムでは、映像（Ｖｉｄｅｏ）、音声（Ａｕｄｉｏ）、ユーザ（Ｕｓｅｒ）等のデータを、ＭＰＥＧ２システム準拠の伝送（ＴＳ）／蓄積（ＰＳ）メディアへリアルタイムで多重（ＭＵＸ）／分離（ＤＭＵＸ）する必要がある。多重化処理では、映像／音声／ユーザのエレメンタリーストリーム（ＥＳ）について、それぞれ専用のパケット（ＰＥＳ）化を図り、それらをトランスポートストリーム（ＴＳ）等に多重化する。この際、時刻管理のためのＰＣＲ、環境情報のＰＳＩ、情報を含まないＮｕｌｌパケット等も多重化する。
【００１１】図９は、従来のＭＰＥＧ２システムにおけるトランスポートストリームＴＳの多重化処理の動作を示すフローチャートである。
【００１２】図９において、基本ポーリングループ内に含まれる各オリジナルタスク（ボックス）は、リアルタイム処理され、各オリジナルタスクが作成する各パケットの多重化の割合は、各エレメンタリーストリームＥＳの規定レートを保つ必要がある。たとえば、Ｖｉｄｅｏ：５．０Ｍｂｐｓ，Ａｕｄｉｏ：２５６ｋｂｐｓ，Ｕｓｅｒ：６４ｋｂｐｓ，ＰＣＲ：１００ｍｓｅｃに１個、ＰＳＩ：１ｓｅｃに１個、全体のＴＳのレートは６．１４４Ｍｂｐｓを保つ必要がある。
【００１３】上記基本ポーリングループ内で繰り返し起動される各オリジナルタスクの実行時間（実行サイク

ル）は、バッファ量やヘッダ検索結果に依存し、数サイクル～数百サイクルと異なる。

【００１４】一方、「トランスポートストリームＴＳの１パケット（１ＴＳ）」が、一定時間以内に処理しなければならない単位であるとすると、各オリジナルタスクをリアルタイム処理するためには、トランスポートストリームＴＳの１パケットを処理する間に、各オリジナルタスクをそれぞれ少なくとも１回起動する必要がある。しかし、このように各オリジナルタスクを単純に起動しようとすると、図１０に示すように、各オリジナルタスクを１回づつ起動する基本ポーリングループを、１つのトランスポートストリームＴＳの時間（一定時間）内に少なくとも１回起動することが不可能である場合（基本ポーリングループを起動すると、一定時間を越える場合）があり、この場合には、基本ポーリングループを、一定時間内に少なくとも１回起動することを保証できないという問題がある。

【００１５】エンベディドシステム上で、従来のリアルタイム情報処理を実現するときに生じる上記ハードウェア、ソフトウェアの問題は、今後開発される種々のエンベディドシステムにおいても同様に生じる問題であり、「高速性」と「柔軟性」とを両立させたエンベディドシステムにおける「オンボード」から「オンチップ」への発展を考慮すると、今後ますます深刻な問題となる。

【００１６】ところが、上記従来例は、上記文献［２］、［３］に示す計算機システム上に、文献［１］に示す割り込みを前提としたリアルタイム情報処理方法を実現するに留まり、現状では、上記問題すら指摘されておらず、勿論これらを解決する手法も提案されていない。

【００１７】本発明は、リアルタイム情報処理において、従来例に不可欠な多重割り込みのプライオリティ制御用のハードウェアと、オリジナルタスク切替えに必要なコンテキストスイッチ用のメモリと、処理時間と等のオーバヘッドを排除することができるリルタイム情報処理方法およびその装置を提供することを目的とするものである。

【００１８】また、本発明は、ＣＰＵと専用ハードウェアとで構成されるエンベディドシステム(Embedded System) 上に、または、それらを１つのチップ上に集積したオンチップ・エンベディドシステム(On Chip Embedded System) 上に、リアルタイム情報処理を構築することができるリアルタイム情報処理方法およびその装置を提供することを目的とするものである。

【００１９】

【課題を解決するための手段】本発明は、複数のオリジナルタスクを、１つのＣＰＵ上で時間多重で処理する場合、上記各オリジナルタスクを、自らのオリジナルタスクよりも小さい複数のサブタスクに分割し、この分割された複数のサブタスクからその１つを選択し、実行するものであり、各オリジナルタスク毎に上記サブタスクを１回づつ起動する基本ポーリングループの起動回数が、所定の一定時間以内に少なくとも１回になるように、上記分割が行われるものである。

【００２０】

【発明の実施の形態および実施例】図１は、本発明の一実施例であるリアルタイム情報処理装置ＲＩＤ１を示す図である。

【００２１】リアルタイム情報処理装置ＲＩＤ１は、ＣＰＵ（計算機）１０と、専用ハードウェア２０とで構成され、ＣＰＵ１０と専用ハードウェア２０との間で情報の授受を行いながら、複数のオリジナルタスクを、ＣＰＵ１０上で時間多重でリアルタイムに処理する装置である。

【００２２】専用ハードウェア２０は、バッファメモリ１と、ライト制御部２と、リード制御部３と、データメモリ４と、複数（m個）の入力ポート７と、複数（n個）の出力ポート８とを有する。

【００２３】バッファメモリ１は、２ポートメモリで構成され、入出力ストリームの一部を格納するバッファメモリである。ライト制御部２は、複数の入力ストリームをバッファメモリ１の第１のポートへ書き込みする動作を制御するライト制御部であり、リード制御部３は、バッファメモリ１の第２のポートから複数の出力ストリームへ読み出しする動作を制御するものである。データメモリ４は、２ポートメモリで構成され、シンタックス処理のために作業データを一時的に格納するメモリである。

【００２４】ＣＰＵ１０は、バッファメモリ１内のデータを参照し、読み出し、データメモリを用いてシンタックス処理を実現し、シンタックス処理されたデータをバッファメモリ１に書き込む処理を実現するプログラム可能な制御部である。

【００２５】また、ＣＰＵ１０は、オリジナルタスク分割手段、サブタスク選択・実行手段の例である。オリジナルタスク分割手段は、上記各オリジナルタスク（処理の主体）を、自らのオリジナルタスクよりも小さい複数のサブタスクに分割する手段である。サブタスク選択・実行手段は、上記分割された複数のサブタスクからその１つを選択し、実行する手段である。なお、上記各オリジナルタスク毎に上記サブタスクを１回づつ起動する基本ポーリングループの起動回数が、所定の一定時間以内に少なくとも１回になるように、上記分割が行われる。

【００２６】オリジナルタスク分割手段と、サブタスク選択・実行手段とによって、基本ポーリングループを１回起動する場合の最大実行サイクルを抑制する。

【００２７】リアルタイム情報処理装置ＲＩＤ１は、メモリ分散化による並列処理を導入している。つまり、バッファメモリ１とデータメモリ４とを分離することによって、ストリーム入出力処理とシンタックス処理とを並

列化している。また、バッファメモリ１とデータメモリ４とを２ポート化することによって、ハードウェアとソフトウェアとからのアクセスを並列化し、ストリームの入力処理と出力処理とを並列化し、これによって、入力側から出力側への不要な転送を排除する装置である。

【００２８】図１に示すリアルタイム情報処理装置ＲＩＤ１において、その横線の破線の上と下とは互いに独立に動作する。つまり、バッファメモリ１とライト制御部２とリード制御部３とで構成される組と、データメモリ４とＣＰＵ１０とで構成される組とは、互いに独立的に動作する。

【００２９】図２は、上記実施例におけるＭＰＥＧ２準拠のシステムパートが規定するプロトコル処理の概要を示す図である。

【００３０】比較のために、ソフトウェアによる実現方式の場合の動作概要も示してある。ソフトウェアで実現した場合、図中の各処理がソフトウェアで処理されるために長時間を要し、さらに、その各処理が逐次的に動作するため、膨大な時間が必要となる。ところが、上記実施例では、各処理部の専用ハードウェア化とそれらの並列パイプライン動作とによって、極めて高速に動作することができる。

【００３１】リード制御部３、ライト制御部２が専用ハードウェア化されていることによって、全体を専用ハードウェア化したときにおける高速の入出力制御と同じ高速性を実現している。また、２ポート化されたバッファメモリ１を用いることによって、ソフトウェアによるシンタックス処理と、専用ハードウェアによる入出力の処理とを並列して動作させることができる。また２ポート化されたデータメモリ４を用いることによって、バッファメモリ１からのデータメモリ１への転送、逆に、データメモリ４からバッファメモリ１へのデータ転送等も並列化される。

【００３２】さらに、入出力ストリーム用のバッファメモリ１を共通化することによって、バッファメモリ１内の入力ストリームをバッファメモリ１内の出力ストリームへ転送する動作を不要化している。たとえば、ＭＰＥＧ２システムパートの多重化（ＭＵＸ）処理においては、入力ストリームの殆どは、ヘッダ類等を付加し、出力ストリームに含まれている。すなわち、入出力バッファを分離すると、この間の不要な転送が頻発する。これらの一連の動作によって、全体として、図２に示したような並列パイプライン処理が可能になる。全体をソフトウェアで処理する方式と比較して、上記実施例では、数十倍以上の高速化を達成できる。

【００３３】次に、ＭＰＥＧ２システムパートの多重化処理を例にとって、上記実施例の全体の動作概要をより詳細に説明する。この場合、入力ストリームとして、映像、音声、ユーザの３つのストリーム（エレメンタリストリーム）を例にとり、出力ストリームとして、トランスポートの場合を例にとる。

【００３４】ライト制御部２（専用ハードウェア）は、入力ポート７から、映像、音声、ユーザデータを入力し、この入力した各データを、バッファメモリ１の異なるアドレス空間上に格納する。この場合、各データのバッファ内アドレス（開始と終了のアドレス）と、所定のヘッダとを検索し、その存在アドレスをＣＰＵ１０に通知する。

【００３５】ＣＰＵ１０は、ライト制御部２からの通知と上記エレメンタリストリームの参照とに従って、各エレメンタリストリームの所定のヘッダ類を作成し、バッファメモリ１の所定のアドレス空間に書き込み、ヘッダ類の長さをリード制御部３に通知する。なお、上記第２の実施例においては、ＣＰＵ１０は、実際には、そのＣＰＵ＋ソフトウェア（プログラム）によって、シンタックス処理制御が実行される。

【００３６】リード制御部３（専用ハードウェア）は、ＣＰＵ１０からの通知に従って、バッファメモリ１内のヘッダ類と所定のパケット長さからヘッダ類の長さを差し引いた分の所定のエレメンタリストリームとを、出力ポート８に送出する。バッファメモリ１は、ライト制御部２、リード制御部３、ＣＰＵ１０の動作に従って、入出力ストリームの一部を格納する。データメモリ４は、ＣＰＵ１０における一時的な作業データを格納する。

【００３７】上記各処理部は、上記動作を繰り返し実行し、ＭＰＥＧ２システムパートが規定する入力ストリームの多重化処理を実現し、出力ストリームを送出する。

【００３８】図３は、上記実施例において、ＭＰＥＧ２システムパートが規定する２つの処理（通信メディア用のトランスポートストリーム処理、蓄積メディア用のプログラムストリーム処理）を実現するときに使用するバッファメモリ１の使用例について示す図である。

【００３９】トランスポートストリーム処理とプログラムストリーム処理との違いは、ＭＰＥＧ２システムパートに規定されているが、それらのシンタックスは、それぞれの使用目的（通信／蓄積）に応じて異なっている。ここで重要なことは、入出力ストリームの入出力タイミングが同一であれば、バッファメモリ１の使用方法（メモリマップ）を変更するだけ、同一のハードウェアによって、異なるシンタックスのプロトコル処理を実現できることである。すなわち、リアルタイム情報処理装置ＲＩＤ１は、シンタックスの変更にも容易に対処可能な柔軟性を有している。

【００４０】上記のように、リアルタイム情報処理装置ＲＩＤ１は、図２に示す動作の高速性と、図３に示す柔軟性とが可能になる。つまり、リアルタイム情報処理装置ＲＩＤ１は、所定のプロトコルに要求される高速性と、所定のプロトコルの変更等へ容易に対処可能な柔軟性とを兼ね備える。すなわち、専用ハードウェアによる高速性と、ソフトウェアによる柔軟性とを両立すること

ができる。これによって、ＭＰＥＧ２システムパートのような複雑で、まだ流動的で、また、リアルタイム処理が不可欠な高速なプロトコルを容易に実現することができる。

【００４１】次に、上記実施例において、従来必要な多重割り込みのプライオリティ制御用のハードウェアと、オリジナルタスク切替えに必要なコンテキストスイッチ用のメモリと、処理時間と等のオーバヘッドを排除することができる点について説明する。

【００４２】次に、上記実施例において、オリジナルタスクＴｏを複数のサブタスクＴｓに分割し、１つのサブタスクＴｓを選択し、実行する動作について説明する。

【００４３】図４（２）は、上記実施例において、オリジナルタスクＴｏを複数のサブタスクＴｓに分割し、その１つのサブタスクＴｓを選択し、実行する動作を示すフローチャートである。

【００４４】図５は、タスクを実行するイメージを示す図であり、図５（１）は、従来の実行イメージを示す図であり、オリジナルタスクＴｏが分割されていないので、図４（１）に示すように、当該オリジナルタスクであれば（Ｓ１１）、ｐａｒｔ１、ｐａｒｔ２、……、ｐａｒｔｎが連続して実行される（Ｓ１１ａ）。図５（２）は、上記実施例において、サブタスクＴｓ毎に実行するイメージを示す図であり、ｎ個のサブタスクＴｓに分割され、ｐａｒｔ１、ｐａｒｔ２、……、ｐａｒｔｎのそれぞれが独立して実行される。

【００４５】図４（２）において、まず、現在起動すべきサブタスクＴｓがｐａｒｔ１（サブタスクＴｓのうちで最初のサブタスクであれば（Ｓ１）、そのｐａｒｔ１を実行し（Ｓ１ａ）、現在起動すべきサブタスクＴｓがｐａｒｔ２（サブタスクＴｓのうちで２番目のサブタスク）であれば（Ｓ２）、そのｐａｒｔ２を実行し（Ｓ２ａ）、……、現在起動すべきサブタスクＴｓがｐａｒｔｎ（サブタスクＴｓのうちでｎ番目のサブタスク）であれば（Ｓｎ）、そのｐａｒｔｎを実行する（Ｓｎａ）。そして、次に実行すべきサブタスクＴｓを設定する制御を行う（Ｓ１０）。上記実施例においては、分割動作と選択動作とが同時に行われているが、分割動作と選択動作とを時間を違えて実行するようにしてもよい。

【００４６】なお、図４において、「ｐａｒｔ＿ｆｌａｇ＿ｎ」は、オリジナルタスクＴｏから分割された複数のサブタスクＴｓのうちでｎ番目のサブタスクＴｓであることを示す変数である。つまり、オリジナルタスクＴｏは、変数「ｐａｒｔ＿ｆｌａｇ＿ｎ」で示されるｎ個のサブタスクに分割され、「ｐａｒｔ＿ｆｌａｇ＿ｎ」の示す部分だけがＣ言語の「ｉｆ＿ｔｈｅｎ」コンストラクトで選択され、実行される。

【００４７】ここで、変数「ｐａｒｔ＿ｆｌａｇ＿ｎ」の値は、排他的に制御される。すなわち、オリジナルタスクＴｏが１回起動される毎に、分割されたｎ個のサブタスクＴｓのうちの１個のサブタスクＴｓのみが選択され、実行されるように制御される。所定のオリジナルタスクＴｏが起動される毎に、上記所定のオリジナルタスクＴｏから分割された複数のサブタスクＴｓのうちで、異なるサブタスクＴｓが順次、選択される。たとえば、１番目のサブタスクＴｓ（Ｐａｒｔ１）→２番目のサブタスクＴｓ（Ｐａｒｔ２）→……→ｎ番目のサブタスクＴｓ（Ｐａｒｔｎ）→１番目のサブタスクＴｓ（Ｐａｒｔ１）→２番目のサブタスク２ＴＳ（Ｐａｒｔ２）→……の順で選択される。なお、各サブタスクＴｓ間で共有される変数はグローバルに扱う必要がある。つまり、各サブタスクＴｓ間で共有すべき変数は、実行すべき対象のサブタスクＴｓが変わっても、常に使用できるようにしてあることが必要である。

【００４８】上記一連の動作（Ｓ１～Ｓ１０）を最初のオリジナルタスクＴｏ（たとえばＶｉｄｅｏ）について実行し、これが終了したら、次のオリジナルタスクＴｏ（たとえばＡｕｄｉｏ）について実行し、……、最後のオリジナルタスクＴｏ（たとえばＰＳＩ）について実行し、この一連の動作（基本ポーリングループ）を繰り返す。

【００４９】上記実施例において、各サブタスクＴｓ（Ｐａｒｔ１，Ｐａｒｔ２，…，Ｐａｒｔｎ）は、オリジナルタスクＴｏを任意の場所で分割されたものであり、オリジナルタスクＴｏから分割されたサブタスクＴｓは、それを実行する場合、各サブタスクＴｓの個別の「ｃａｌｌ」から「ｒｅｔｕｒｎ」まで実行され、サブタスクＴｓの実行サイクルがオリジナルタスクＴｏの実行サイクルよりも短い。すなわち、オリジナルタスクＴｏを分割した後におけるオリジナルタスクの実行サイクルは、オリジナルタスクＴｏの実行サイクルよりも短い。しかも、所定のオリジナルタスクＴｏが分割されたサブタスクＴｓの実行サイクルを、その分割のし方によって、任意に設定することができ、したがって、１つの基本ポーリングループ（各オリジナルタスク毎にサブタスクを１回づつ起動するループ）においては上記所定のオリジナルタスクＴｏが部分的にしか実行されないが、上記所定のオリジナルタスクＴｏの実行サイクルを実質的に任意の大きさに合わせることができる。なお、オリジナルタスクＴｏが任意の大きさに分割され、実行されても、細切れではあるが、一連の処理が実行され、オリジナルタスクの実行という点では問題が生じない。

【００５０】図６は、上記実施例における基本ポーリングループ起動回数設定の説明図である。

【００５１】まず、各オリジナルタスク毎にサブタスクＴｓを１回づつ起動する基本ポーリングループの起動回数N_{BPL} を、一定時間T_{const} 以内に所定の一定回数以上に設定する。上記実施例における一定時間T_{const} は、上記実施例をＭＰＥＧ２システムに適用した場合において、リアルタイム処理しなければならない一定時間

であり、この一定時間の単位として、トランスポートストリームＴＳの１パケットを処理する時間（１トランスポートストリームＴＳ）を設定してある。

【００５２】また、基本ポーリングループＢＰＬは、複数のオリジナルタスクＴｏ（たとえばＶｉｄｅｏ、Ａｕｄｉｏ、Ｕｓｅｒ、ＰＣＲ、ＰＳＩの各オリジナルタスク）のそれぞれのサブタスクＴｓによって構成され、順次ポーリングされるループであり、この基本ポーリングループが１回呼び出されると、各オリジナルタスクＴｏが、それぞれ部分的ではあるが、少なくとも１回起動されることを図６で示してある。

【００５３】基本ポーリングループの起動回数N_{BPL}は、一定時間内（１ＴＳ）に、基本ポーリングループが起動する回数であり、一定時間内（１ＴＳ）に起動される基本ポーリングループの起動回数N_{BPL}は固定ではないが、オリジナルタスク分割手段によってオリジナルタスクＴｏをサブタスクＴｓに分割する分割方法に応じて、一定時間内に設定される起動回数N_{BPL}が定まる。

【００５４】Ｖｉｄｅｏタスクを分割したサブタスクのうちで最も長い時間T_{Vmax}と、Ａｕｄｉｏタスクを分割したサブタスクのうちで最も長い時間T_{Amax}と、Ｕｓｅｒタスクを分割したサブタスクのうちで最も長い時間T_{Umax}と、ＰＣＲタスクを分割したサブタスクのうちで最も長い時間T_{Pmax}とが１つの基本ポーリングループに含まれていたとしてもこれらの時間の総和も、一定時間内（１ＴＳ）である。

【００５５】図６に示すように、一定時間内（１ＴＳ）に基本ポーリングループＢＰＬが少なくとも１回起動されることによって、一定時間内（１ＴＳ）に各オリジナルタスクが少なくとも１回確実に起動されている。これに対して、上記従来方法では、一定時間内（１ＴＳ）に基本ポーリングループＢＰＬが少なくとも１回起動されることが困難な場合があり、この場合には、一定時間内（１ＴＳ）に各オリジナルタスクが少なくとも１回起動させることを保証することができない。

【００５６】図７は、上記実施例をＭＰＥＧ２システムに適用した場合の一例の説明図であり、基本ポーリングループＢＰＬを起動する動作の概要を示す図である。

【００５７】ここでは、一定時間内（１ＴＳ）にリアルタイム処理をしなければならない単位として、トランスポートストリームＴＳの１パケットの処理時間を設定してある。１トランスポートストリームＴＳ内のポーリングのタイミングは、たとえば任意周期の螺旋曲線のように刻々と変化する。しかし、上記オリジナルタスク分割手段と、サブタスク選択・実行手段と、基本ポーリングループ起動回数設定手段とによって、基本ポーリングループＢＰＬに含まれる各オリジナルタスクが、一定時間内（１トランスポートストリームＴＳ）にN_{BPL}回以上スケジュールされることが保証される。

【００５８】図８は、上記実施例をＭＰＥＧ２システムに適用した場合の具体的な例を示す図である。

【００５９】図８には、実際のトランスポートストリームの処理において、Ｖｉｄｅｏ、Ａｕｄｉｏ、Ｕｓｅｒ、ＰＣＲ、ＰＳＩ、Ｎｕｌｌ等の起動すべき各オリジナルタスクによって生成される各パケットの順序や割合を示してある。この場合、各エレメンタリーストリームＥＳの実際の規定のレートは、Ｖｉｄｅｏ：５．０Ｍｂｐｓ，Ａｕｄｉｏ：２５６ｋｂｐｓ，Ｕｓｅｒ：６４ｋｂｐｓ，ＰＣＲ：１００ｍｓｅｃに１個、ＰＳＩ：１ｓｅｃに１個、全体のトランスポートストリームＴＳのレートは６．１４４Ｍｂｐｓであり、図８に示す各オリジナルタスクによって作成される各パケットの割合は、これらの規定レートに従ったものになっていることがわかる。

【００６０】上記実施例によれば、従来のリアルタイム情報処理方法で不可欠な割り込みを前提とせず、ポーリングを前提とした処理方法であるので、従来のリアルタイム情報処理方法で特に問題となっている多重割り込みのプライオリティ制御用の複雑なハードウェアを必要とせず、また、オリジナルタスク切替えに必要なコンテキストスイッチ用の膨大なメモリと処理時間とを必要とせず、また、従来のリアルタイム情報処理装置よりも小型であり、経済的である。

【００６１】また、上記実施例のキーポイントは、リアルタイム情報処理の実現における割り込み処理の不要化である。ＣＰＵとメモリを中心とした通常の計算機システムや、オンボードやオンチップのエンベディドシステム上にリアルタイム情報処理を今後とも構築していく必要があり、複雑な割り込み制御は、これらのシステムの高速化を阻止する要因の１つであるので、割り込みを不要化した上記実施例の意義は大きい。また、ＣＰＵとメモリとを中心としたリアルタイム情報処理システムは、ＶＬＳＩ技術の進展、特に、メモリの高速化、大容量化と、ＣＰＵの高速化等の恩恵を直接受けることができ、さらに、リアルタイム情報処理システムの簡単化と汎用化等、将来的に有望な情報処理システムの構成を可能とする新しい情報処理方法である。

【００６２】さらに、ＣＰＵとメモリとを中心としたシステムの構成は、ＶＬＳＩ上にインプリメントするに際しても、ＶＬＳＩ技術の進展によって、今日では、高速なオンチップの内蔵型ＣＰＵ（コアＣＰＵと呼ぶ）や、高速大容量なオンチップのメモリ等がＶＬＳＩライブラリとして供給されているので、１チップのＶＬＳＩ化が容易に可能になる。したがって、上記実施例は、容易にＶＬＳＩ化することができるシステムであり、このＶＬＳＩ化によって、従来のリアルタイム情報処理方法に比べ、小型化、経済化を図ることができる。

【００６３】ＭＰＥＧ２システムパートプロトコル処理を実現する場合、ハードウェアとして、図１に示す装置を適用し、そのソフトウェア（ファームウェア）とし

て、上記実施例を利用すると、オンチップ・エンベディドシステムを容易に実現できるので、１チップのＶＬＳＩ化が可能であり、従来のリアルタイム処理を実現するＭＰＥＧ２のシステムパートのプロトコル処理装置に比べて、小型であり経済的であるリアルタイム情報処理可能なＭＰＥＧ２システムパートのプロトコル処理装置を実現することができる。

【００６４】また、上記実施例では、リアルタイム情報処理として、それを実現するためのハードウェアへの無理な要求がなく、極めて簡単な情報処理システムを提供することができる。ＶＬＳＩ上にインプリンメントする場合、コアＣＰＵやメモリ等はライブラリとして提供可能なので、上記実施例を実現する場合、アプリケーションに必要な機能のみを専用ハードウェアで実現するだけで足りる。これらのアプリケーションに必要な機能は、近年の論理合成技術で極めて容易に実現できる。このように、上記実施例は、ライブラリ化された（部品化された）コンポーネントをベースとした極めて簡単な構成であるので、ＶＬＳＩ設計技術における上位合成技術（たとえばシステムレベルシンセシス等）の対象のターゲットアーキテクチャとしても極めて有効であり、上位合成技術の恩恵を受け、近い将来、上記実施例をベースにしたリアルタイム情報処理システムを上位仕様記述に応じて、ＶＬＳＩ上に自動合成することが可能になる。

【００６５】上記のように、将来的なネットワーク技術をベースにしたマルチメディア時代において、上記実施例によって提供される小型、経済化、ＶＬＳＩ化が可能な新しいリアルタイム情報処理装置の果たす役割は図り知れない。したがって、上記実施例は、ＣＰＵと専用ハードウェアとで構成されるエンベディドシステム(Embedded System) 上に、または、それらを１つのチップ上に集積したオンチップ・エンベディドシステム(On Chip Embedded System) 上に、リアルタイム情報処理を構築するに好適である。

【００６６】なお、上記実施例において、オリジナルタスクＴｏをサブタスクＴｓへ分割する方法、その分割されたサブタスクＴｓを選択的に実行させるための変数の使用の方法等は、任意に定めることができる。また、ターゲットのシステムの構成、通常の計算機システムまたはエンベディドシステム、オンボードまたはオンチップ等を、任意に定めることができる。さらに、上記実施例において、ＭＰＥＧ２システムパートのプロトコル処理を用いたが、その応用の種類等も、任意に定めることができる。また、上記実施例において、専用ハードウェア２０の代わりに他の専用ハードウェアを使用するようにしてもよい。

【００６７】なお、上記実施例を、１つのチップ上に構成し、所定のリアルタイム情報処理を実現するようにしてもよい。

【００６８】さらに、上記実施例を方法の発明として把握することができ、つまり、上記実施例は、１つのＣＰＵと専用ハードウェアとで構成され、上記ＣＰＵと上記専用ハードウェアとの間で情報の授受を行いながら処理を進める情報処理装置によって、複数のオリジナルタスクを、上記ＣＰＵ上で時間多重で処理するリアルタイム情報処理方法において、上記各オリジナルタスクを、自らのオリジナルタスクよりも小さい複数のサブタスクに分割するオリジナルタスク分割段階と、上記分割された複数のサブタスクからその１つを選択し、実行するサブタスク選択・実行段階とを有し、上記各オリジナルタスク毎に上記サブタスクを１回づつ起動する基本ポーリングループの起動回数が、所定の一定時間以内に少なくとも１回になるように、上記分割が行われ、所定のリアルタイム情報処理を実現することを特徴とするリアルタイム情報処理方法として把握することができる。

【００６９】また、上記実施例を専用ハードウェアを使用しない方法に適用することができる。つまり、複数のオリジナルタスクを１つのＣＰＵ上で時間多重で処理するリアルタイム情報処理方法において、上記各オリジナルタスクを、自らのオリジナルタスクよりも小さい複数のサブタスクに分割するオリジナルタスク分割段階と、上記分割された複数のサブタスクからその１つを選択し、実行するサブタスク選択・実行段階とを有し、上記各オリジナルタスク毎に上記サブタスクを１回づつ起動する基本ポーリングループの起動回数が、所定の一定時間以内に少なくとも１回になるように、上記分割が行われ、所定のリアルタイム情報処理を実現することを特徴とするリアルタイム情報処理方法として把握することができる。

【００７０】また、上記実施例を、専用ハードウェアを使用しない装置、つまり、複数のオリジナルタスクを、１つのＣＰＵ上で時間多重で処理するリアルタイム情報処理装置に適用することができる。

【００７１】

【発明の効果】本発明によれば、リアルタイム情報処理において、従来、不可欠な多重割り込みのプライオリティ制御用のハードウェアと、オリジナルタスク切替えに必要なコンテキストスイッチ用のメモリと処理時間と等のオーバヘッドを排除することができるという効果を奏する。

【図面の簡単な説明】

【図１】本発明の一実施例のリアルタイム情報処理装置ＲＩＤ１を示す図である。

【図２】上記実施例において、ＭＰＥＧ２準拠のシステムパートが規定するプロトコル処理の概要を示す図である。

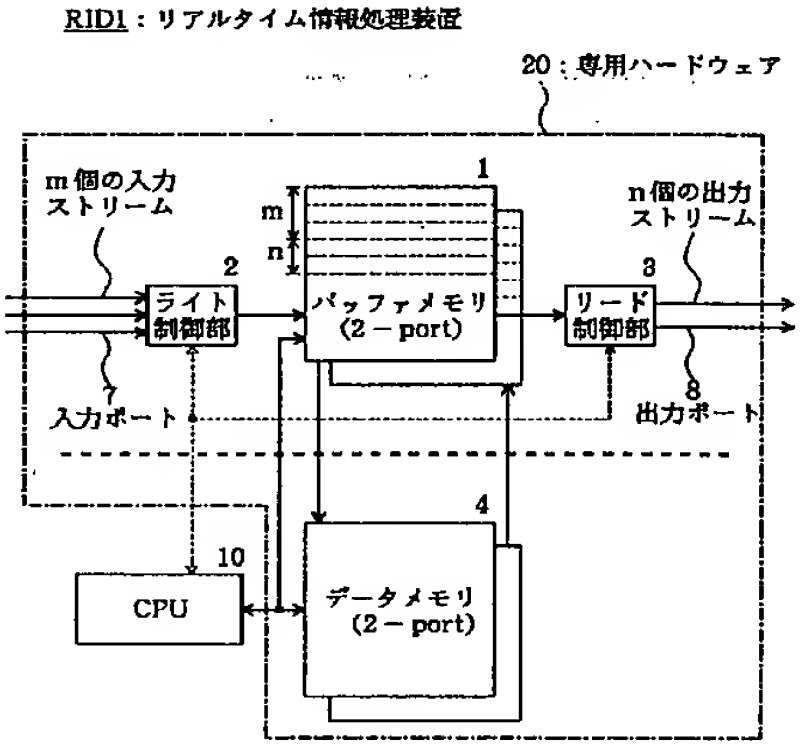
【図３】上記実施例において、ＭＰＥＧ２システムパートが規定する２つの処理を実現するときに使用するバッファメモリ１の使用例について示す図である。

【図４】オリジナルタスクＴｏまたはサブタスクＴｓを

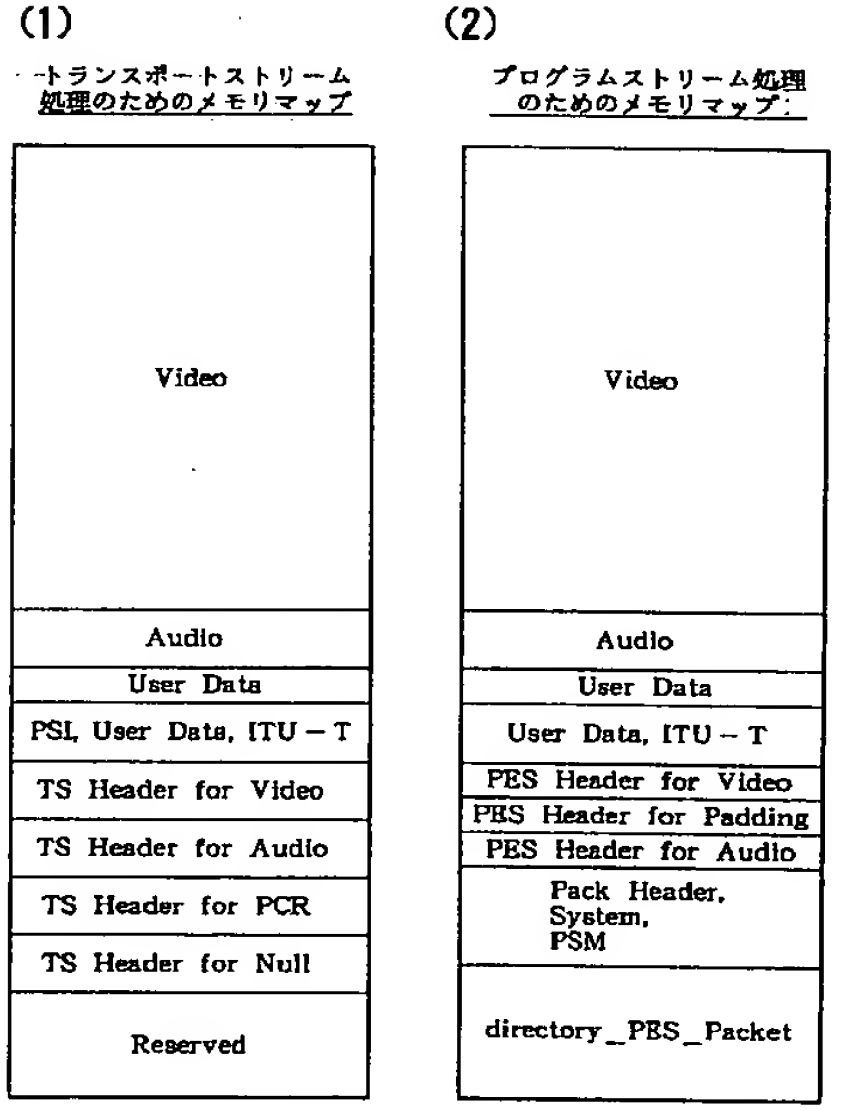
実行する動作を示すフローチャートである。
【図５】タスクを実行するイメージを示す図である。
【図６】上記実施例における基本ポーリングループ起動回数設定の説明図である。
【図７】上記実施例をＭＰＥＧ２システムに適用した場合の一例の説明図であり、基本ポーリングループＢＰＬを起動する動作の概要を示す図である。
【図８】上記実施例をＭＰＥＧ２システムに適用した場合の具体例を示す図である。
【図９】従来のＭＰＥＧ２システムにおけるトランスポートストリームＴＳの多重化処理の動作を示すフローチャートである。

【図１０】従来例において、各オリジナルタスクを起動する場合にリアルタイム処理が不可であることを示す図である。
【符号の説明】
ＲＩＤ１…リアルタイム情報処理装置、
１０…ＣＰＵ、
２０…専用ハードウェア、
Ｔｏ…オリジナルタスク、
Ｔｓ…サブタスク、
ＢＰＬ…基本ポーリングループ、
N_{BPL} …基本ポーリングループの起動回数。

【図１】



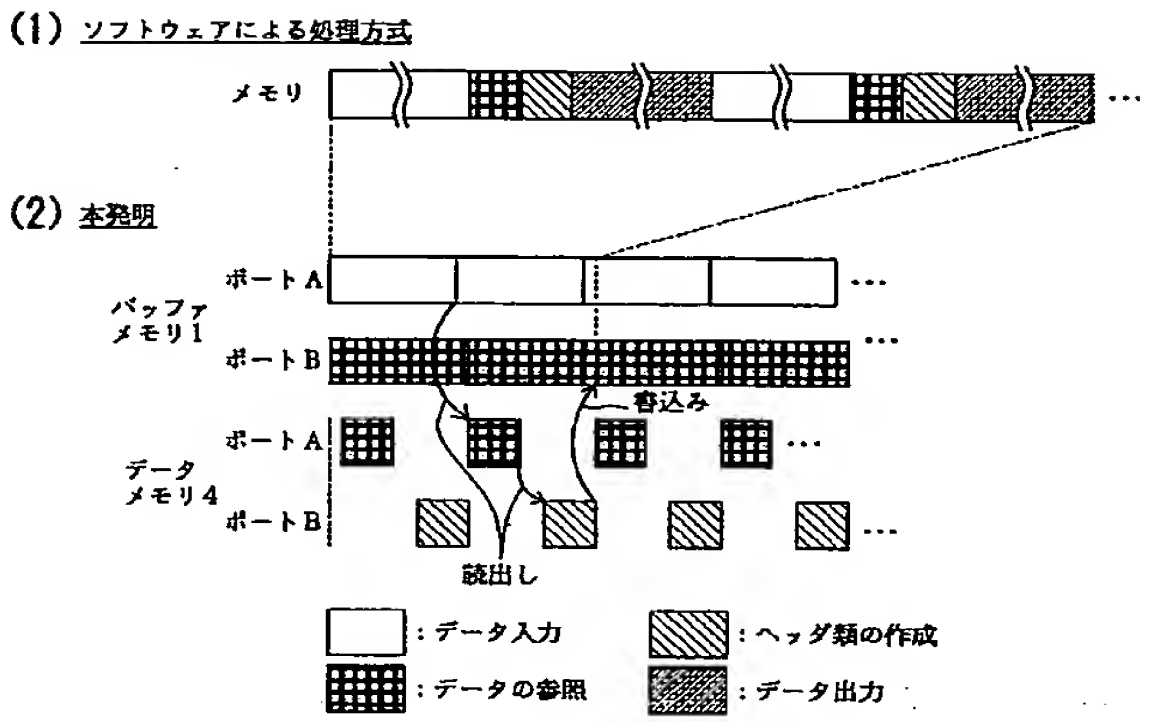
【図３】



K3900

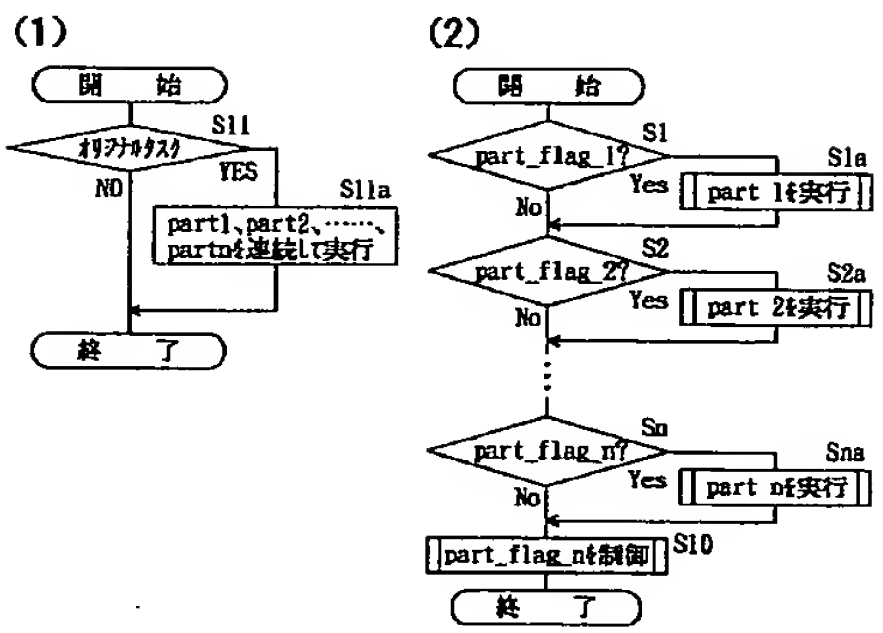
K3900

【図２】

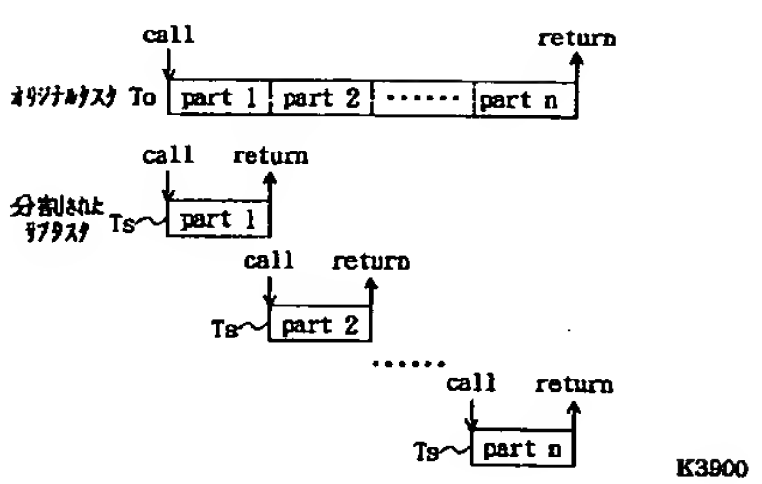


K3900

【図４】

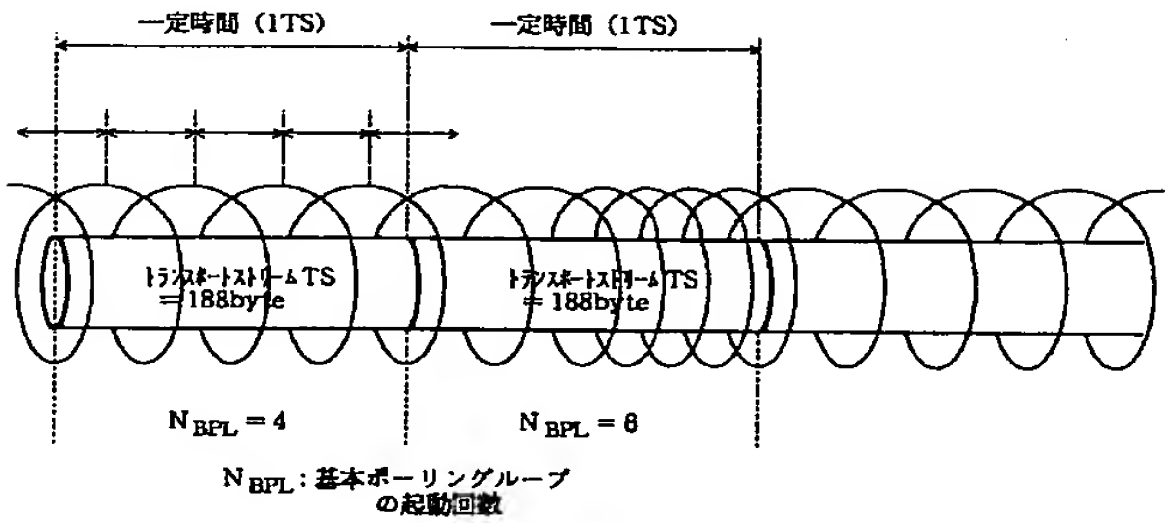


【図５】



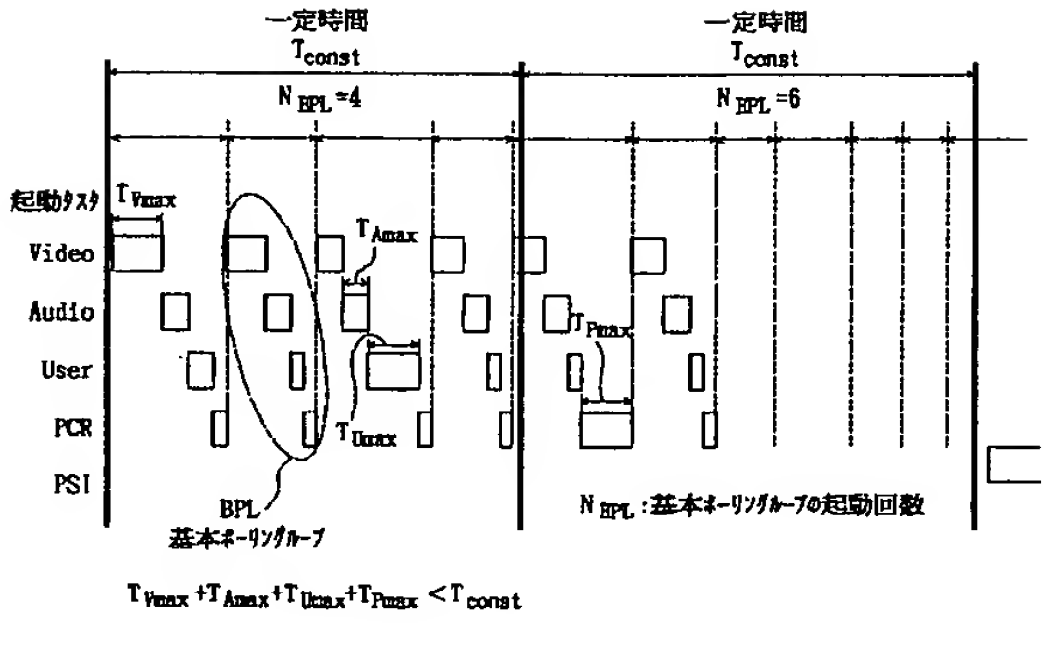
K3900

【図７】

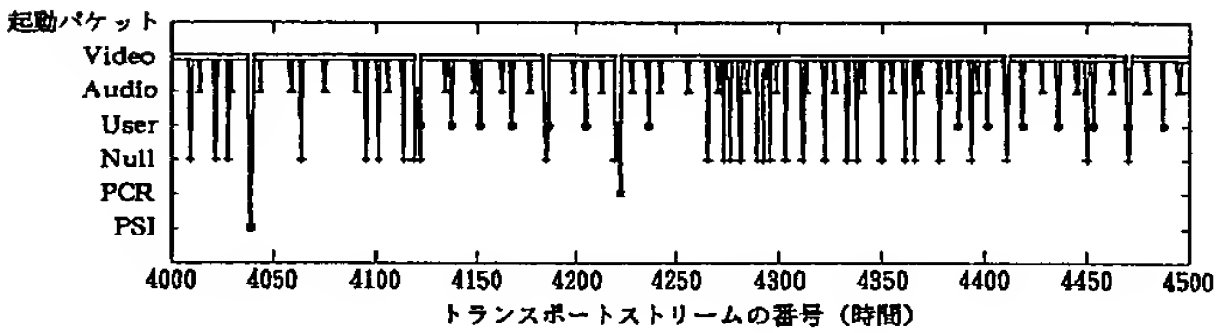


K3900

【図６】

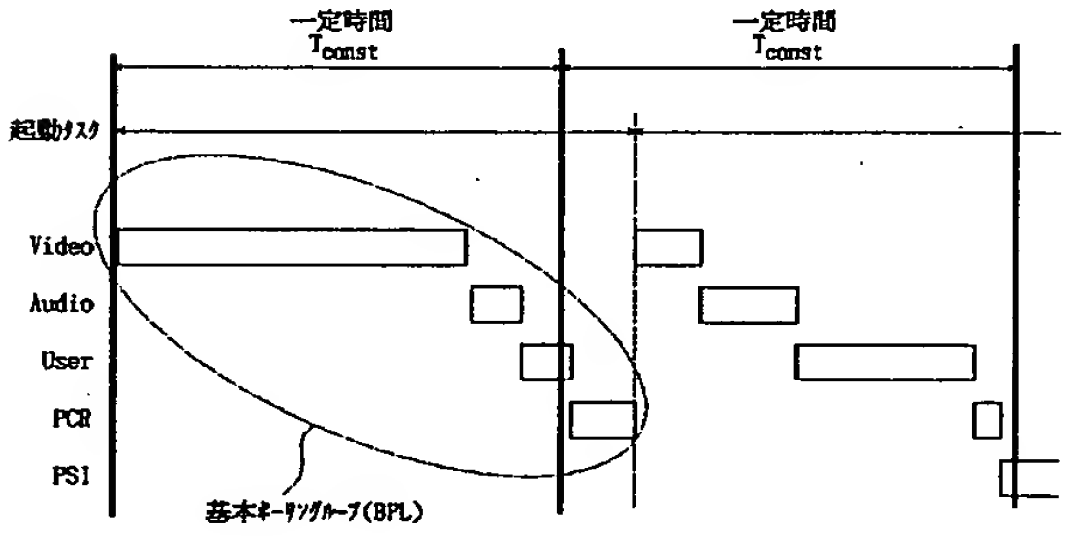


【図８】



【図１０】

従来例における各タスクの起動状態（リアルタイムの処理は不可）



【図９】

従来例におけるMPEG2のTSの多重化処理の動作